WHAT IS CLAIMED IS:

1. An apparatus for providing transparent fault tolerance within an application server environment comprising a network of computers, said apparatus comprising:

   a. a first server designated as a master server for storing and operating a first operating system program communicating by system calls with a first server application program and a first fail over protection program, said first server designated as a master server connected to a computer network and having a network address; said first server having a first initial state, a first application state and a first network connection state;

   b. a second server designated as a back-up server for storing and operating a second operating system program communicating by system calls with a second server application program and a second fail over protection program; said second operating system program, said second server application program and said second fail over protection program identical respectively to said first operating system program, said first server application program and said first fail over protection program; said second server designated as a back-up server connected to said computer network; said second server having a second initial state, a second application state and a second network connection state

   c. wherein the first server designated as a master server is operatively connected to the second server designated as a back-up server and wherein the first server is in continuous communication with said second server so that the first fail over protection program is in constant communication with the second fail over protection program and further wherein the operation of the first server and second server are synchronized by the first and second fail over protection programs respectively;

   d. wherein the first and second fail over protection programs include:

i. means for establishing synchronicity between the first server and the second server;

ii. means for monitoring synchronicity between the first server and the second server;

iii. means for detecting non-synchronicity between the first server and the second server; and,

iv. means for invoking the first or second fail over protection programs upon detection of non-synchronicity between the first and second servers;

e. wherein said first and second fail over protection programs, when invoked, cause a transfer of server operations from a failed server to a non-failed server upon the detection of non-synchronicity or non-responsiveness of either server, and wherein transfer from failed to non-failed server is totally transparent to the client.

2. The apparatus as claimed in claim 1, wherein means for establishing synchronicity between the first server and the second server includes means for:

a. synchronizing the first and second initial state;

b. synchronizing the first and second application state; and,

c. synchronizing the first and second network connection state.

3. The apparatus as claimed in claim 2 wherein means for synchronizing the first and second application states includes means for intercepting system calls between the first server application program and the first operating program.

4. A method for providing transparent fault tolerance within an application server environment comprising a network of computers, said method comprising the steps of:

a. providing a first server for storing and operating a first operating system program, a first server application program and a first fail over protection program;

b. providing a second server for storing and operating a second operating system program, a second server application program and a second fail over protection program;

c. placing said first server in continuous communication with said second server;

d. designating from the first server and the second server a master server and a back-up server;

e. synchronizing the operation of the master server and the back-up server;

f. providing from the network an identical client data stream input simultaneously to the master server and the back-up server wherein:

i. the master server and back-up server have the same network address

ii. the master server and back-up server simultaneously process said identical client data stream; and wherein,

iii. the master server and the back-up server simultaneously produce a respective first and second output data streams; and wherein,

iv. said first and said second output data streams are identical if the master server and the back-up server are operating correctly;

g. comparing by said first and second fail over protection programs respectively, said first output data stream with said second output data stream for divergence from identicality of the first output data stream from the second output data stream;

h. detecting by said first and second fail over protection programs no divergence from identicality of the first output data stream from the second output data stream;

5. The method of claim 4 including the steps of:

a. receiving by said first or second fail over protection programs an indication of divergence from identicality of the first output data stream from the second output data stream;

b. invoking the first or second fail over protection program wherein the backup server assumes the duty of the master server without breaking any network connections.

6. The method as claimed in claim 5, wherein the first and second operating system programs and the first and second server application programs are deterministic so that when the first and second operating system programs and the first and second server application programs receive the same input they will produce the same output.

7. The method as claimed in claim 6 wherein the step of synchronizing the first master and second back-up servers comprises the steps of:

a. providing to each of the master and back-up operating system programs identical executables, configuration files and data files prior to starting the master and back-up operating system programs;

b. synchronizing the operation of the master application server program with the back-up application server program so that the master and back-up application server programs have an identical internal operating state and so that each of the master

and back-up application server programs produce an identical first and second data output respectively; and,

c. synchronizing the network connection state between the master server and back-up server application programs and the network.

8. The method as claimed in claim 7, wherein synchronization of the master and back-up server application programs comprises the steps of:

a. providing the master server and the back-up server with identical interfaces to the network;

b. providing in each of the master and back-up servers a system call interceptor which will intercept system calls traveling from their respective server application systems to their respective operating system programs;

c. starting the master and the back-up server application programs; and,

d. synchronizing the result of system calls between master and backup.

9. The method as claimed in claim 8, wherein synchronizing the network connection state between the network and the master and back-up server application programs comprises the following steps:

a. providing identical network addresses to the master and back-up servers;

b. providing a simulated network layer within the master server and back-up servers;

c. providing a client data stream to each of the master server and back-up server;

d. receiving said client data stream by the master server simulated network layer;

e. transmitting the client data stream received by the master server simulated network layer to the master server application program;

f. processing the client data stream by the master server application program;

g. detecting differences in the master and backup's output; and,

h. invoking the first fail over protection program.